

**Amendments to the Claims:**

1-24 (Cancelled).

25 (New): A computer-implemented method for providing a common operating system, the method comprising:

identifying components of a monolithic operating system, wherein the operating system has a monolithic structure and components are arranged in a hierarchical relationship;

identifying dependencies between the components of the monolithic operating system, wherein the dependencies are determined based on the position of the dependencies within the monolithic operating system;

generating component groups from the identified dependencies, wherein the component groups share common dependencies;

generating dependency rules, wherein the dependency rules are configured to provide fewer dependencies than the dependencies in the monolithic operating system; and

enforcing the dependency rules to establish the common operating system, wherein the common operating system is configured to support the components of the generated component groups.

26 (New): The computer-implemented method of claim 25, wherein generating component groups from the identified dependencies includes generating subset headers for components of the component group, wherein the subset headers eliminate dependencies on components in upper layers of the hierarchical relationship of the monolithic operating system.

27 (New): The computer-implemented method of claim 25, wherein the common operating system utilizes less memory than the monolithic operating system.

28 (New): The computer-implemented method of claim 25, further comprising testing one of the components of the groups of components to validate reliability of the component.

29 (New): The computer-implemented method of claim 25, further comprising testing component manifests based on the common operating system.

30 (New): The computer-implemented method of claim 25, wherein the common operating system is configured as a pre-boot operating system.

31 (New): The computer-implemented method of claim 25, further comprising accessing the common operating system for componentized testing.

32 (New): The computer-implemented method of claim 25, wherein the component groups are generated based on the function of the components of the monolithic operating system.

33 (New): A computer readable storage medium having computer executable instructions for providing a common operating system, the instructions comprising:

identifying components of a monolithic operating system, wherein the operating system has a monolithic structure and components are arranged in a hierarchical relationship;

identifying dependencies between the components of the monolithic operating system, wherein the dependencies are determined based on the position of the dependencies within the monolithic operating system;

generating component groups from the identified dependencies, wherein the component groups share common dependencies, wherein the components of the component groups include subset headers, wherein the subset headers eliminate dependencies on components in upper layers of the hierarchical relationship of the monolithic operating system;

generating dependency rules, wherein the dependency rules are configured to provide fewer dependencies than the dependencies in the monolithic operating system; and

enforcing the dependency rules to establish the common operating system, wherein the common operating system is configured to support the components of the generated component groups.

34 (New): The computer-readable storage medium of claim 33, wherein the common operating system utilizes less memory than the monolithic operating system.

35 (New): The computer-implemented method of claim 33, further comprising testing one of the components of the groups of components to validate reliability of the component.

36 (New): The computer-implemented method of claim 33, further comprising testing component manifests based on the common operating system.

37 (New): The computer-implemented method of claim 33, wherein the common operating system is configured as a pre-boot operating system.

38 (New): The computer-implemented method of claim 33, further comprising accessing the common operating system for componentized testing.

39 (New): The computer-implemented method of claim 33, wherein the component groups are generated based on the function of the components of the monolithic operating system.

40 (New): A system for providing a common operating system, the system comprising:

a processor; and

a memory having computer executable instructions stored thereon for:

identifying components of a monolithic operating system, wherein the operating system has a monolithic structure and components are arranged in a hierarchical relationship;

identifying dependencies between the components of the monolithic operating system, wherein the dependencies are determined based on the position of the dependencies within the monolithic operating system;

generating component groups from the identified function of each component, wherein the component groups share common dependencies, wherein the components of the component groups include subset headers, wherein the subset headers eliminate dependencies on components in upper layers of the hierarchical relationship of the monolithic operating system;

generating dependency rules, wherein the dependency rules are configured to provide fewer dependencies than the dependencies in the monolithic operating system; and

enforcing the dependency rules to establish the common operating system, wherein the common operating system is configured to support the components of the generated component groups.

41 (New): The system of claim 40, wherein the common operating system utilizes less memory than the monolithic operating system.

42 (New): The system of claim 40, further comprising testing one of the components of the groups of components to validate reliability of the component.

43 (New): The system of claim 40, further comprising testing component manifests base on the common operating system.

44 (New): The system of claim 40, further comprising accessing the common operating system for componentized testing.